Thrilled by Your Progress! Large Language Models (GPT-4) No Longer Struggle to Pass Assessments in Higher Education Programming Courses

Year: 2023 | Citations: 128 | Authors: Jaromir Savelka, Arav Agarwal, Marshall An, C. Bogart, M. Sakr

---

Abstract

This paper studies recent developments in large language models' (LLM) abilities to pass assessments in introductory and intermediate Python programming courses at the postsecondary level. The emergence of ChatGPT resulted in heated debates of its potential uses (e.g., exercise generation, code explanation) as well as misuses in programming classes (e.g., cheating). Recent studies show that while the technology performs surprisingly well on diverse sets of assessment instruments employed in typical programming classes the performance is usually not sufficient to pass the courses. The release of GPT-4 largely emphasized notable improvements in the capabilities related to handling assessments originally designed for human test-takers. This study is the necessary analysis in the context of this ongoing transition towards mature generative AI systems. Specifically, we report the performance of GPT-4, comparing it to the previous generations of GPT models, on three Python courses with assessments ranging from simple multiple-choice questions (no code involved) to complex programming projects with code bases distributed into multiple files (599 exercises overall). Additionally, we analyze the assessments that were not handled well by GPT-4 to understand the current limitations of the model, as well as its capabilities to leverage feedback provided by an auto-grader. We found that the GPT models evolved from completely failing the typical programming class' assessments (the original GPT-3) to confidently passing the courses with no human involvement (GPT-4). While we identified certain limitations in GPT-4's handling of MCQs and coding exercises, the rate of improvement across the recent generations of GPT models strongly suggests their potential to handle almost any type of assessment widely used in higher education programming courses. These findings could be leveraged by educators and institutions to adapt the design of programming assessments as well as to fuel the necessary discussions into how programming classes should be updated to reflect the recent technological developments. This study provides evidence that programming instructors need to prepare for a world in which there is an easy-to-use widely accessible technology that can be utilized by learners to collect passing scores, with no effort whatsoever, on what today counts as viable programming knowledge and skills assessments.